

# 2017년 프로그래밍언어 개론

-03-

제출일자	2017.03.28.
이 름	정 윤 수
학 번	201302482
분 반	03

**def \_\_init\_\_(self, type, lexeme):**

```
def __init__(self, type, lexeme):
    """
    :type type:CuteType
    :type lexeme: str
    :param type:
    :param lexeme:
    :return:
    """
    # Initialize the token to generate.
    # Fill Out
    if check_keyword(lexeme) :
        self.type = _get_keyword_type(lexeme)
    else:
        self.type = type
    self.lexeme = lexeme

    print "[" + CUTETYPE_NAMES[self.type] + ": " + self.lexeme + "]"
```

## 문제 해결 방법

init()함수는 Token클래스의 객체가 생성이 되어질 때 호출이 되어지는 함수이다 이 함수는 매개변수로 type과 lexeme를 받는데 lexeme에는 단어가 들어가 있고 type에는 lexeme로 들어온 값은 DFA에서 어느 상태에 해당을 하는지에 대한 정보를 저장을 하고 있다. 먼저 init()함수의 변수에 매개변수로 들어온 값들을 저장을 하기 전에 현재 들어온 단어가 특수한 문자인지 확인을 해야한다. 그러기 위해서는 check\_keyword()함수를 이용을 하여서 특수문자인지 판별을 한다 만약 현재 들어온 단어가 특수한 문자라면 \_get\_keyword\_type()함수를 이용을 하여 self.type 변수에 반환값을 저장을 하고 특수문자가 아니라면 매개변수로 들어온 type의 값을 self.type에 저장해주고 매개변수로 들어온 lexeme의 값을 self.lexeme에 저장을 해주고 print()함수를 이용을 하여 단어의 상태 타입과 단어를 출력을 해준다.

**def \_get\_node(self, index, x):**

```
def _init_TM(self):
    for alpha in letters:
        self.transM[(0, alpha)] = 4
        self.transM[(4, alpha)] = 4

    for digit in digits:
        self.transM[(0, digit)] = 1
        self.transM[(1, digit)] = 1
        self.transM[(2, digit)] = 1
        self.transM[(4, digit)] = 4
    self.transM[4, '?'] = 16
    self.transM[0, ')'] = 6
    self.transM[0, '-'] = 2
    self.transM[0, '+'] = 3
    self.transM[0, '('] = 5
    self.transM[0, '#'] = 7
    self.transM[7, 'T'] = 8
    self.transM[7, 'F'] = 9
    self.transM[0, '*'] = 10
    self.transM[0, '/'] = 11
    self.transM[0, '^'] = 15
    self.transM[0, '>'] = 13
    self.transM[0, '='] = 14
    self.transM[0, '<'] = 12
```

## 문제 해결 방법

init\_TM() 함수는 transition matrix를 정의를 해주는 함수이다. 맨 처음 for문에서는 모든 문자들에 대해서 상태 전이를 정의를 해주고 그 다음 for문에서는 모든 숫자에 대해서 상태전이를 정의를 해준다. 그 다음 나머지 전이들은 DFA에서의 상태전이를 보고 정의를 해주었다.

## 느낀점

이번 과제를 해보면서 처음에는 이 프로그램이 어떻게 동작을 이해를 하고 값들이 어떻게 출력이 되는지 먼저 확인을 한 후 프로그래밍을 시작을 하였다. 이번 과제에서는 transition matrix를 이용을 하여 각 단어마다 한 글자씩 살펴보면서 DFA를 따라 상태전이를 하는 것을 할 수 있었다. 또한 특수한 키워드들을 따로 처리하는 것 또한 볼 수있었고 함수를 하나 하나씩 살펴 보아서 DFA 상태전이를 프로그래밍으로 구현을 하는 것을 좀 더 잘 알수있었다.